

虚拟校园系统设计说明书

09023324 万佰鑫

09023315 李仕轩

09023227 金耿贤

09023228 张圣勤

09023304 刘 硕

版本：1.0-demo

日期：2025 年 8 月 29 日

修改日期	版本号	修改人	修改描述
2025 年 8 月 29 日	0.0.1-demo	万佰鑫、金耿贤	奠基

目录

1	引言	4
1.1	编写目的	4
1.2	背景	4
1.3	定义	4
1.4	参考资料	4
1.4.1	项目内部文件	4
1.4.2	国家及行业标准	4
1.4.3	技术与架构设计文献	5
2	程序系统的分析	5
2.1	可行性分析	5
2.2	需求分析	6
2.2.1	功能需求	6
2.2.2	非功能需求	6
2.3	开发设计环境	6
2.3.1	IntelliJ IDEA	6
2.3.2	JDK 17	7
2.3.3	MySQL 8.4.6 LTS	7
2.3.4	Java FX	7
2.3.5	Maven	8
3	程序系统的结构	8
4	公共模块设计说明	9
4.1	消息信息模型	9
4.1.1	请求结构Request	9
4.1.2	响应结构Response	9
4.2	用户信息模型	9
4.3	工具类	9
4.3.1	密码工具类	9
4.3.2	数据库工具类DbHelper	9
5	网络模块设计说明	9
5.1	Socket	10
5.1.1	客户端实现	10
5.1.2	服务端NettyServer实现	10
5.2	输入输出流	10
6	多线程模块设计说明	11
6.1	客户端	11
6.2	服务器端	11

7 数据库设计说明	11
7.1 元数据	11
7.2 表设计	11
7.2.1 user表	11
7.2.2 student表	11
7.2.3 course表	11
7.2.4 class表	11
7.2.5 book表	11
7.2.6 finance_card表	11
7.2.7 select_record表	11
7.2.8 teaching_evaluation表	11
7.2.9 store_item表	11
7.2.10 store_transaction表	11
7.2.11 library_transaction表	11
7.2.12 card_transaction表	11
8 分工表	11
9 进度计划表	12

1 引言

1.1 编写目的

本文档旨在详细阐述“虚拟校园 (VCampus)”项目的整体设计方案，明确其系统架构、模块划分与核心技术细节。

该系统旨在构建一个多功能的虚拟校园平台，核心功能模块包括：用户管理（含注册、登录与授权）、学生学籍管理、在线选课系统、数字图书馆、校园商店、线上银行以及一个集成的LLM智能聊天界面。

为确保系统的高效、稳定与可扩展性，本次设计遵循了一系列现代软件工程原则。系统整体采用客户端/服务器（C/S）架构，并通过Socket通信实现数据交互。在业务实现上，我们遵循严格的分层设计（Layered Architecture）与模型-视图-控制器（MVC）模式，以实现业务逻辑与用户界面的有效分离。此外，项目还将采用分布式开发思想，通过多线程处理并发用户请求，为项目构建一个结构清晰、易于协作和维护的开发基础。

本文档是指导项目开发全过程的文件。开发团队严格遵循本文档所定义的架构、接口与规范进行编码、测试与集成工作，以确保精确实现上述各模块功能的过程中高效协作。并且本文档为学生、老师、管理员等用户提供针对该项目详实的使用说明。

1.2 背景

软件系统名称: 虚拟校园 (VCampus) 系统。项目任务来源: 本项目是“暑期学校专业技能实训课程”设定的课程设计任务，旨在通过构建一个功能完善的软件系统，锻炼和检验开发团队在需求分析、架构设计、团队协作及项目管理方面的综合能力。开发者: 姓名目标用户: 本系统的主要用户群体为虚拟校园环境中的在校学生、教职员工以及负责后台维护的系统管理员。运行环境: 系统基于客户端/服务器 (C/S) 架构，服务器端程序需部署在安装了兼容 JDK 17 版本 Java 运行环境的服务器上，并连接至 MySQL 数据库。客户端程序为桌面应用程序，可在主流个人计算机操作系统（如 Windows, macOS, Linux）上跨平台运行。

1.3 定义

系统 指本文档所描述的虚拟校园系统软件。

用户 包括学生、教师和系统管理员。

1.4 参考资料

1.4.1 项目内部文件

《暑期学校专业技能实训课程安排》：本文档是项目的任务来源规定了项目的总体要求、功能模块、技术栈约束及考核标准。

《软件设计说明书DEMO》：为本次设计说明书的编写提供了内容结构和格式参考。

1.4.2 国家及行业标准

GB/T 8567-2006 计算机软件文档编制规范：本设计说明书的结构和内容参考了该国家标准，以确保文档的规范性。

Google Java Style Guide: 作为项目编码规范的主要参考,用于统一团队的代码风格、命名约定和格式化标准,以提升代码的可读性和可维护性。

Oracle Code Conventions for the Java(TM) Programming Language: 官方Java编程语言代码约定,作为编码规范的补充参考。

1.4.3 技术与架构设计文献

《Effective Java》(3rd Edition) by Joshua Bloch: Java 语言的权威实践指南,为编写出更清晰、健壮、可复用的高质量Java代码提供了宝贵的建议。

《设计模式:可复用面向对象软件的基础》(Design Patterns: Elements of Reusable Object-Oriented Software): 经典的 GoF 设计模式参考手册,为项目中如 MVC、单例等模式的应用提供了理论指导。

《代码整洁之道》(Clean Code: A Handbook of Agile Software Craftsmanship): 软件工程领域的经典著作,指导开发者编写易于理解、易于维护的代码,其思想贯穿于本项目的编码实践中。

《架构整洁之道》(Clean Architecture: A Craftsman's Guide to Software Structure and Design): 深入阐述了软件架构设计的核心原则,如组件、层次、边界和依赖关系等,是本项目进行分层设计和模块解耦的重要理论指导。

2 程序系统的分析

2.1 可行性分析

本节旨在对“虚拟校园 (VCampus)”项目的可行性进行综合评估,从技术实现、经济效益和实际操作三个维度,判断项目是否具备开发条件。

从技术可行性角度看,本项目是完全可实现的。项目要求采用客户端/服务器(C/S)架构,并通过Java Socket及多线程技术进行开发,这些都是成熟且拥有丰富文档支持的主流技术。开发环境(Eclipse、兼容JDK 1.7的Java版本)和数据库(Access/MySQL)均为标准、易于获取的工具,不存在技术壁垒。虽然团队成员主要以C++为开发语言,但具备扎实的计算机科学基础,将现有编程知识迁移至Java语言,并掌握Swing、Socket等API不存在根本性困难。因此,在技术层面,本项目风险可控。

在经济可行性方面,我们需将常规的“投资回报率”调整为“投入产出比”来评估。本项目的主要“投入”是开发团队在课程期间投入的时间和精力,而非大量的资金。所需的开发软件均为开源或免费版本,硬件则基于团队成员的个人计算机,几乎无直接的财务成本。而项目的“产出”则具有极高的价值:它不仅是完成“暑期学校专业技能实训课程”的必要条件,更是一次宝贵的、完整的软件工程实践。通过本项目,团队成员将获得在分层架构设计、网络通信、数据库交互以及团队协作方面的综合实战经验,这对未来的职业发展是重要的无形资产。因此,本项目具备显著的经济可行性。

最后,就操作可行性而言,VCampus系统旨在将校园中分散的线下业务(如学籍管理、选课、图书馆借阅等)整合到统一的线上平台,这符合教育信息化的大趋势。对于目标用户(学生、教师、管理员),该系统能显著简化办事流程,提高信息获取效率。系统所模拟的业务流程均源于真实的校园场景,易于用户理解和接受。只要用户界面设计得足够直观易用,系统在理论上的校园环境中推广和使用,不存在明显的操作障碍。

综上所述,VCampus项目在技术、经济和操作层面均被证明是可行的。项目团队具备完成开发所需的技术学习和实践能力,项目成果能带来远超投入的知识回报,且其产品定位符合实际应用场景的需求。因此,项目具有可行性。

2.2 需求分析

本章节将详细定义VCampus系统的功能性需求与非功能性需求，作为后续系统设计、开发、测试及验收的基准。

2.2.1 功能需求

功能需求详细描述了系统应为用户提供的各项服务和功能。

1. 用户管理模块（此模块是系统的基础，为所有用户提供身份认证与管理服务）注册: 新用户（学生或管理员）能够通过界面输入必要信息（如用户名、密码、角色）创建新账户。登录/登出: 已注册用户可以通过唯一的用户名和密码登录系统，访问其权限对应的功能。用户也可以随时安全退出登录状态。授权: 系统需根据用户角色（学生、管理员）区分权限，确保不同角色只能访问其被授权的功能模块。
2. 学生学籍管理模块（该模块主要面向管理员，用于管理学生的基本信息）信息录入与修改: 管理员可以添加、删除和修改学生的学籍信息。信息查询: 支持通过学号、姓名等关键信息快速查询学生资料。列表视图: 通过列表查看所有学生的信息。
3. 选课系统模块（此模块为学生提供在线选课服务，是系统的核心业务功能之一）课程信息浏览: 学生可以查看当前学期所有可选课程的列表，包括课程名称、授课教师、学分、学时等信息。在线选课/退课: 在规定时间内，学生可以根据个人培养计划选择或退选课程。个人课表查询: 学生可以查看自己已选课程的时间和地点安排。
4. 图书馆模块（模拟校园图书馆的核心功能）图书查询: 用户可以根据书名、作者等信息检索馆藏图书。借阅/归还: 模拟图书的借阅和归还流程。图书信息浏览: 可以通过列表视图浏览图书信息。
5. 校园商店与银行模块（提供虚拟的校园生活服务）商店: 用户可以浏览商品并使用虚拟货币进行购买。银行: 提供虚拟货币的存取款、转账等基本银行服务。
6. LLM聊天界面模块（集成一个基于大语言模型的智能问答界面，为用户提供校园信息咨询等服务）

2.2.2 非功能需求

非功能需求定义了系统的质量属性和在运行环境中必须遵守的约束。

性能需求: 系统应具备良好的响应能力。在高并发场景下（例如选课高峰期），服务器端需能通过多线程机制有效处理多个客户端的同时请求，确保系统不出现严重延迟或无响应。

兼容性需求: 系统开发必须严格遵守技术栈约束。后端逻辑需使用Java语言编写，JDK版本必须兼容1.7。数据库可选用Access或MySQL，但需保证SQL语句的通用性。

可靠性与健壮性: 客户端与服务器端的Socket通信必须建立完善的异常处理机制，能够处理网络中断、超时等常见问题，避免程序因网络波动而崩溃。

安全性需求: 用户密码等敏感信息在数据库中存储时不应为明文。客户端与服务器之间传输的所有对象都必须实现可序列化接口，并保持版本一致性，以防止数据传输错误。

代码规范与可维护性: 项目代码必须遵循统一的编码规范（如Google Java Style Guide），并为主要的类和方法提供标准的JavaDoc注释。系统的设计应严格遵循分层原则，确保各层之间职责清晰、耦合度低，便于后续的功能扩展与维护。

2.3 开发设计环境

2.3.1 IntelliJ IDEA

本项目计划采用IntelliJ IDEA作为核心集成开发环境（IDE）。该工具由JetBrains开发，广泛应用于Java项目，其突出优势体现在智能代码补全、动态错误检测、强大的重构功能以及内建的版本控制系统集成，在java编程方面具有较大的优势。IntelliJ IDEA所提供的卓越编辑辅助功能能够显著加速开发流程，其与Maven的深度集成也确保了项目构建过程的稳定性和可重复性。

IntelliJ IDEA在此项目之中所体现的核心优势体现在以下几个方面：

1.智能编码辅助与动态分析：IntelliJ IDEA提供强大的智能代码补全、动态错误检测和即时代码质量分析功能。这些功能能够实时识别潜在的代码缺陷和不规范之处，并提供优化建议。这不仅加快了编码速度，也从源头保障了代码的健壮性和可维护性。

2.强大的重构功能：对于多人协作的“V-campus”项目而言，代码结构会随着开发进程不断演进。IntelliJ IDEA提供的强大、安全的自动化重构功能（如类更名、方法提取等），能够确保在修改项目结构时，关联引用能够被正确更新，从而有效维护了代码库的完整性。

3.深度集成版本控制系统：该IDE内置了对Git、Subversion等主流版本控制系统的深度支持。开发人员无需离开IDE环境即可完成代码提交、分支管理、冲突解决等版本控制操作，极大地简化了团队协作流程。

此外，IntelliJ IDEA也对Maven等构建工具提供了深度支持，能够有效管理项目依赖与构建流程，显著提升开发效率与代码可维护性。

2.3.2 JDK 17

本项目的核心开发语言环境选用Java Development Kit (JDK) 17。该版本由Oracle指定为长期支持（LTS）版本，这代表着它将获得长达数年的稳定安全、性能和错误修复更新。JDK 17在语言特性、运行性能及安全性方面具备多项改进，如引入了Sealed Classes、Pattern Matching for Switch等新语法特性，增强了类型系统的表达力与代码的简洁性。其优化的垃圾回收机制和提升的编译效率也为系统的高并发处理与稳定运行提供了坚实基础。同时该版本兼容性较强，也更加稳定。选择LTS版本有助于为“虚拟校园”系统的长期稳定运行提供坚实保障，避免了因使用非LTS版本而可能面临的频繁版本升级和兼容性风险。

2.3.3 MySQL 8.4.6 LTS

数据库管理系统方面本软件设计采用MySQL 8.4.6 LTS版本。该版本不仅完全支持ACID事务，还提供诸如窗口函数、通用表表达式（CTE）和JSON字段增强功能等高级特性，极大丰富了数据操作与查询的能力。

此外，MySQL 8.4.6在身份验证、数据加密与权限管理等方面引入多项安全机制，保障了数据的机密性与完整性，非常适用于对数据一致性要求较高的校园管理系统。此版本不仅完全支持ACID事务，确保了在处理选课、学生学籍变更等核心业务中数据的一致性与完整性，还提供了注入公共表达式以及窗口函数等高级特性，极大丰富了数据操作能力。

2.3.4 Java FX

图形用户界面（GUI）部分基于Java FX框架实现。作为一个现代、跨平台的UI框架，Java FX提供了丰富的可视化组件、灵活的布局控制（FXML）以及CSS样式定制能力，能够构建响应迅速、视觉效果良好的桌面应用程序。其与JDK的紧密集成简化了部署过程，同时具备与底层图形硬件的良好协同，显著提升了客户端的用户体验。其核心优势首先体现在现代化与跨平台，这代表了JavaFX提供了丰富的可视化组件和灵活的布局控制，能够构建响应迅速、视觉效果良好的桌面应用程序。其跨平台特性确保了客户端系统在不同操作系统上（如Windows, macOS, Linux）表现出一致的用户体验。

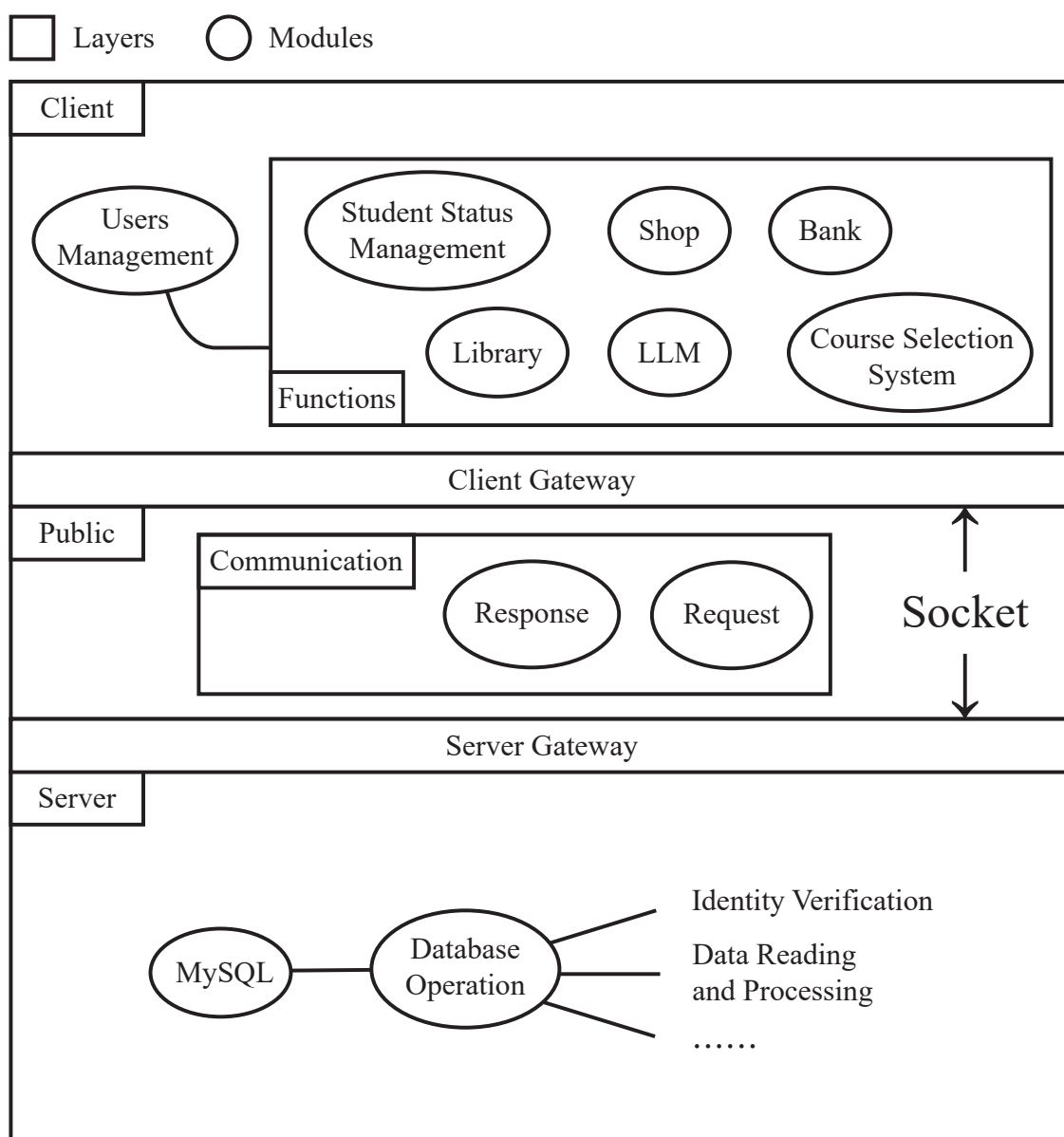
其次体现在了设计与逻辑分离，JavaFX支持使用FXML（一种基于XML的声明式UI语言）来定义界面布局，并使用CSS样式表来定制界面风格。这种模式将界面设计与后端业务逻辑进行了清晰的分离，完全契合了项目中强调的“分层设计”原则（MVC）。它允许团队中的UI设计师与后端开发人员独立工作，从而显著提升开发效率。

2.3.5 Maven

本项目还计划采用Apache Maven作为项目构建与依赖管理工具。Maven是一款基于项目对象模型（POM）的软件项目管理工具，可自动化完成项目的编译、测试、打包及部署等生命周期管理任务。通过规范的目录结构和统一的依赖管理机制，Maven显著提升了项目的可维护性与团队协作效率。

其强大的插件生态系统和与主流IDE（如IntelliJ IDEA）的无缝集成，进一步确保了构建过程的可重复性和环境一致性，为大型多模块项目的依赖控制与构建流程提供了稳定基础。

3 程序系统的结构



4 公共模块设计说明

4.1 消息信息模型

4.1.1 请求结构Request

我们计划自定义一个通用的请求格式，而且所有客户端发往服务器的请求都应当遵循这个结构，结构主要包括目标服务路径（uri）、携带的参数（params）以及用于身份验证的会话信息（session）。这样设计的好处是前后端交互格式统一，便于扩展和调试。

4.1.2 响应结构Response

在服务器处理完请求后，会返回一个结构固定的响应，包含操作状态（status）、提示信息（message）、实际返回的数据（data）以及更新后的会话信息（session）。这种结构有助于客户端快速判断请求结果并提取所需数据。

4.2 用户信息模型

系统中所有用户（包括学生、教师、管理员等）的基本信息都通过一个用户模型进行管理，涵盖账号、姓名、联系方式、权限角色等常用字段，为各功能模块提供基础数据支持。

4.3 工具类

工具类核心部分主要分为两块，即为密码工具类以及数据库工具类。

4.3.1 密码工具类

本工具类则负责用户密码的加密和验证。本项目计划采用 BCrypt 算法来对明文密码进行哈希处理，设置成本因子，生成随机盐值后使用EksBlowfish 算法计算生成最终的哈希字符串，同时在程序之中使用verify() 方法用于校验传入的密码是否与存储的哈希值匹配。

4.3.2 数据库工具类DbHelper

这个工具类主要封装了数据库连接及会话管理的基本操作，其提供了一个静态方法 init() 用于初始化数据库连接并返回 Hibernate 的 Session 对象，避免在各处重复编写连接代码。

5 网络模块设计说明

本项目采用 Netty 框架处理底层网络通信，实现了一套基于 Socket 的客户端-服务器架构，它作为VCampus系统客户端与服务器端之间通信的基石，其设计的可靠性、高效性与安全性直接决定了整个系统的运行质量。通信数据使用 JSON 格式进行序列化与反序列化，通过自定义的编解码器处理请求与响应报文，旨在构建一个稳定、可扩展的通信框架，以满足虚拟校园多用户实时交互的业务需求。

5.1 Socket

Socket是网络模块中最基础的通信端点，Java原生Socket API为建立TCP连接提供了直接支持。在本暑期实训项目之中计划设计的V-Campus系统中，Socket的创建与管理策略在客户端和服务端有着明确的区分。

5.1.1 客户端实现

本项目计划将客户端通过封装好的 Socket 客户端类建立与服务器的长连接，支持发送请求并异步接收响应，内部自动处理连接重试和异常断开。

在客户端，Socket的创建发生在其需要与服务器进行交互时。通常，会在应用程序初始化阶段或用户登录时，通过new Socket尝试与指定的服务器地址和端口建立连接。该连接一旦成功建立，便会被缓存并复用于该客户端会话期间后续的所有网络通信，直至用户登出或应用程序关闭时才显式关闭，以避免频繁建立和断开连接所带来的性能损耗。客户端需要负责监控此连接的状态，并在连接意外中断时触发重连机制，以保证服务的连续性。

5.1.2 服务端NettyServer实现

服务器端主要基于Netty-NIO框架构建，负责监听指定端口、接收客户端连接、处理网络I/O事件并分发业务请求。其核心设计与功能如下：

1. 采用主从Reactor线程模型，bossGroup线程组负责接受新连接，workerGroup线程组负责处理已建立连接的I/O操作，有效提升并发处理能力。
2. 通过ServerBootstrap类进行启动配置，初始化通道管道（Pipeline），指定编解码器及业务处理器。
3. 提供run(router, database)方法，接收全局路由管理器（Router）及数据库会话等依赖项后启动服务。

5.2 输入输出流

在本系统中，所有网络通信均基于Netty框架实现，通过自定义的编解码器处理请求与响应数据的序列化与反序列化。输入输出流的处理是网络通信层的核心部分，确保数据能够正确、高效地在客户端与服务器之间传输。

6 多线程模块设计说明

6.1 客户端

6.2 服务器端

7 数据库设计说明

7.1 元数据

7.2 表设计

7.2.1 user表

7.2.2 student表

7.2.3 course表

7.2.4 class表

7.2.5 book表

7.2.6 finance_card表

7.2.7 select_record表

7.2.8 teaching_evaluation表

7.2.9 store_item表

7.2.10 store_transaction表

7.2.11 library_transaction表

7.2.12 card_transaction表

8 分工表

学号	名称	职务	具体工作	联系方式
09023324	万佰鑫	组长	架构 & 文档 & 类声明	414498351
09023315	李仕轩	组员	Maven 项目管理	2746293150
09023227	金耿贤	组员	文档 & 公共模块	736704509
09023228	张圣勤	组员	功能模块	1704139044
09023304	刘 硕	组员	JavaFX 开发	3349318467

9 进度计划表

第一周	8月25日 - 8月29日
进度	已完成分工，初步设计项目的架构、数据库和登录
状态	各模块具体功能和 API 还需讨论
计划	在第二周初步实现数据库通信和各功能模块
建议	合理分工，统筹调度